A truncated nonmonotone Gauss-Newton method for large-scale nonlinearleast-squares problems.

Computational Optimization and Applications

DOI: 10.1007/s10589-006-6444-2

A Truncated Nonmonotone Gauss-Newton Method for Large-Scale Nonlinear Least-Squares Problems

G. FASANO fasano@iasi.cnr.it
Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti" - CNR, Viale Manzoni 30, 00185 Roma, Italy; Istituto Nazionale per Studi ed Esperienze di Architettura Navale INSEAN, Via di Vallerano 139, 00128 Roma, Italy

F. LAMPARIELLO lampariello@iasi.cnr.it
Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti" - CNR, Viale Manzoni 30, 00185 Roma, Italy

M. SCIANDRONE sciandro@iasi.cnr.it
Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti" - CNR, Viale Manzoni 30, 00185 Roma, Italy

Received April 14, 2004; Revised July 13, 2005

Published online: 23 March 2006

Abstract. In this paper, a Gauss-Newton method is proposed for the solution of large-scale nonlinear least-squares problems, by introducing a truncation strategy in the method presented in [9]. First, sufficient conditions are established for ensuring the convergence of an iterative method employing a truncation scheme for computing the search direction, as approximate solution of a Gauss-Newton type equation. Then, a specific truncated Gauss-Newton algorithm is described, whose global convergence is ensured under standard assumptions, together with the superlinear convergence rate in the zero-residual case. The results of a computational experimentation on a set of standard test problems are reported.

Keywords: large-scale problems, nonlinear least-squares, truncated Gauss-Newton method, nonmonotone line search techniques

1. Introduction

We consider the nonlinear least-squares problem

$$\min_{x\in R^n} f(x) = \frac{1}{2}\|r(x)\|^2 = \frac{1}{2}\sum_{i=1}^{m} r_i^2(x), \quad m \geq n,$$

where each component $r_i : R^n \to R$ of the residual vector $r(x)$ is a twice continuously differentiable function, and n is large.

Let $J(x)$ be the Jacobian matrix of $r(x)$. Then, the gradient $\nabla f(x)$ and the Hessian matrix $\nabla^2 f(x)$ are given by

$$\nabla f(x) = J(x)^T r(x), \quad \nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x).$$

Given a point x_k, the Gauss-Newton method determines a new point $x_{k+1} = x_k + d_k$, where d_k is a solution of the linear system

$$J(x_k)^T J(x_k)\, d = -\nabla f(x_k). \tag{1}$$

The Gauss-Newton method exploits the particular structure of $\nabla^2 f(x)$, by discarding the second-order term. It is well-known that the method is only locally convergent to a stationary point $x^\star$, provided that $\nabla^2 f(x^\star)$ is nonsingular and the second-order term is small relative to $J(x^\star)^T J(x^\star)$. Moreover, in the zero-residual case the convergence rate is superlinear. The global convergence of the method can be ensured by applying a line search or a trust region strategy (see, e.g., [5, 13]).

When dealing with large-scale problems, the solution of the linear system (1) may be prohibitively costly, so that it may be convenient to use inexact methods like those underlying truncated-Newton methods. The convergence theory developed for the latter (see, e.g., [3, 4, 12]) can be used as reference to establish that of a truncated version of the Gauss-Newton method. In such a version, Eq. (1) can be solved approximately by applying, for instance, the conjugate gradient method (CG) and by interrupting the iterations according to a suitable rule. In this connection, we observe that the coefficient matrix $J(x_k)^T J(x_k)$ is always positive semidefinite, Eq. (1) admits at least a solution, and all the conjugate vectors generated by CG are of descent for f. Moreover, in order to ensure the convergence of the overall method, the matrix $J(x_k)^T J(x_k)$ may be easily modified, e.g., by simply adding a multiple of the identity matrix. Note that the requirements for the application of the standard CG are satisfied, while in a truncated-Newton method this may not hold, so that it is necessary to take into account that the Hessian may be indefinite.

In this paper we define a truncated Gauss-Newton method for large-scale problems by adopting a truncation strategy in the method described in [9]. The latter was designed with the aim of obtaining a behavior as close as possible to that of the "pure Gauss-Newton" method, i.e., by taking the unit stepsize along the search direction obtained by solving (1). In particular, a modification of the coefficient matrix in the Gauss-Newton equation was introduced only at a subsequence of iterates, and the use of a nonmonotone line search technique allowed us to accept the pure Gauss-Newton iteration more frequently than a standard monotone one.

In Section 2 we establish sufficient conditions for ensuring the convergence of an iterative method employing a truncation scheme for computing the search direction, as approximate solution of a Gauss-Newton type equation. In Section 3 we describe a truncated nonmonotone Gauss-Newton method and we prove its convergence properties. Finally, the numerical results obtained by solving a set of test problems from the literature are compared with those derived by applying a truncated-Newton method, and a standard routine (NAG library) suggested for large-scale problems.

2. A truncation scheme for computing the search direction

For solving the problem $\min_{x \in R^n} f(x)$, we consider an iterative method of the form

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots \tag{2}$$

where d_k is the search direction, and the stepsize α_k along it is determined by means of a suitable line search. We assume that d_k is computed by solving inexactly the equation

$$B_k d = -\nabla f(x_k), \tag{3}$$

where B_k is a symmetric matrix approximating the Hessian $\nabla^2 f(x_k)$, and that Eq. (3) admits a solution, i.e., $\nabla f(x_k)$ belongs to $\mathcal{R}(B_k)$, the range or column space of B_k. Moreover, likewise in [9], we state the following conditions on the minimum and maximum eigenvalues $\lambda_{\min}(B_k)$ and $\lambda_{\max}(B_k)$:

(a) there exists a constant $c > 0$ such that for all k

$$0 \leq \lambda_{\min}(B_k) \leq \lambda_{\max}(B_k) \leq c;$$

(b) let $K_p = \{k_o, k_1, \ldots, k_i, k_{i+1}, \ldots\} \subseteq \{0, 1, \ldots\}$ be a subset of iterates such that $k_{i+1} - k_i \leq p, \forall i$, where p is a prefixed integer; we assume that for every infinite subset $K \subseteq K_p$

$$\lim_{k\to\infty, k\in K} \lambda_{\min}(B_k) = 0 \qquad \text{implies} \qquad \lim_{k\to\infty, k\in K} \|\nabla f(x_k)\| = 0.$$

These conditions essentially ensure that the subsequence $\{d_k\}_{k\in K_p}$ is gradient-related to $\{x_k\}_{k\in K_p}$. Note that the particular choice (periodic) of K_p such that $k_{i+1} - k_i = p, \forall i$, will be adopted later. Then, we consider the following algorithm for computing d_k, i.e., for obtaining an approximate solution of (3).

Truncated Conjugate Gradient Algorithm (TCG)

Data. x_k, $\nabla f(x_k) \neq 0$, B_k, $\eta_k \in (0, 1)$.
Step 0. Set $p_o = 0$, $q_o = s_o = -\nabla f(x_k)$, $i = 0$.
Step 1. Compute

$$\delta_i = \frac{s_i^T q_i}{s_i^T B_k s_i}$$

$$q_{i+1} = q_i - \delta_i B_k s_i$$

$$p_{i+1} = p_i + \delta_i s_i.$$

Step 2. If $\|q_{i+1}\|/\|\nabla f(x_k)\| \leq \eta_k$, set $d_k = p_{i+1}$, $\rho_k = -q_{i+1}$ and exit, else compute

$$\beta_i = \|q_{i+1}\|^2/\|q_i\|^2$$
$$s_{i+1} = q_{i+1} + \beta_i s_i,$$
set $i = i + 1$ and go to Step 1.

Note that the vector ρ_k represents the error in Eq. (3), i.e., we have

$$B_k d_k = -\nabla f(x_k) + \rho_k. \tag{4}$$

We show that Algorithm TCG is well-defined, and that any vector p_i is of descent for f.

Proposition 1. *In Algorithm* TCG*, let B_k be symmetric positive semidefinite and assume that $\nabla f(x_k) \in \mathcal{R}(B_k)$. Then,*

(i) *for all $i \geq 1$, $s_i^T B_k s_i = 0$ if and only if $q_i = 0$;*

(ii) *there exists an integer $h \in [0, n-1]$ such that*

$$\|q_{h+1}\| / \|\nabla f(x_k)\| \leq \eta_k;$$

(iii) $\nabla f(x_k)^T p_i < 0$ *for all $i \geq 1$.*

Proof: We note first that $s_o^T B_k s_o > 0$, since $s_o = -\nabla f(x_k) \neq 0$, and by assumption $s_o \in \mathcal{R}(B_k)$, and thus $s_o \notin \mathcal{N}(B_k)$, the null space of B_k (which is positive semidefinite).

(i) By simple substitutions, we have $q_i^T s_{i-1} = 0$ for all $i \geq 1$, so that,

$$s_i^T q_i = \|q_i\|^2. \tag{5}$$

Then, if $q_i = 0$, we have $\beta_{i-1} = 0$, and hence $s_i = 0$.
On the other hand, since $q_o = s_o = -\nabla f(x_k) \in \mathcal{R}(B_k)$, we have $s_i, q_i \in \mathcal{R}(B_k)$ for all i. Moreover, $s_i^T B_k s_i = 0$ implies that s_i belongs to $\mathcal{N}(B_k)$, so that $s_i \in \mathcal{R}(B_k) \cap \mathcal{N}(B_k) = \{0\}$. Then, from (5), it follows $q_i = 0$.

(ii) Assume by contradiction that $\|q_{h+1}\| > 0$ for all $h \in [0, \ldots, n-1]$, so that, by (i)

$$s_i^T B_k s_i > 0, \qquad \text{for all } i \geq 0. \tag{6}$$

Following the same reasoning employed in [8] for the case that B_k is positive definite, we have

$$q_n^T s_i = 0, \quad \text{for all } i = 0, \ldots, n-1 \tag{7}$$

and

$$s_i^T B_k s_j = 0, \quad \text{for all } i, j = 0, \ldots, n-1, \ i \neq j.$$

The latter implies, together with (6), the linear independence of $s_0, \ldots, s_{n-1}$. Then, by (7), we have $q_n = 0$, which contradicts the assumption.

(iii) Following again the same reasoning employed in [8] for the case that B_k is positive definite, we have $s_i^T q_i = s_i^T q_o$, for all $i \geq 0$, which yields

$$\begin{aligned}\nabla f(x_k)^T p_i &= -\sum_{j=0}^{i-1} \left(s_j^T q_j / s_j^T B_k s_j\right) q_o^T s_j \\ &= -\sum_{j=0}^{i-1} \left(s_j^T q_o\right)^2 / s_j^T B_k s_j \leq -\|q_o\|^4 / s_o^T B_k s_o < 0.\end{aligned}$$

□

We can prove the following convergence result.

Proposition 2. *Let $\{x_k\}$ be the sequence generated by the iterative scheme* (2)*. Assume that B_k is symmetric positive semidefinite , $\nabla f(x_k) \in \mathcal{R}(B_k)$, conditions* (a) *and* (b) *are satisfied, and d_k is computed by means of Algorithm* TCG*, where $\eta_k \to 0$ for $k \to \infty$. Moreover, assume that the stepsize α_k is determined by a line search in such a way that*

$$\lim_{k\to\infty, k\in K_p} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0, \tag{8}$$

and that

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{9}$$

Then, every limit point of $\{x_k\}$ is a stationary point of $f(x)$.

Proof: Let $K \subseteq \{0, 1, \ldots\}$ be any infinite subset for which the subsequence $\{x_k\}_K$ converges, i.e.,

$$\lim_{k\to\infty, k\in K} x_k = \bar{x}.$$

For each $k \in K$, let $\ell(k) \in [0, p-1]$ be the smallest integer such that $k + \ell(k) \in K_p$. Then, since

$$\left\|x_{k+\ell(k)} - x_k\right\| \leq \left\|x_{k+\ell(k)} - x_{k+\ell(k)-1}\right\| + \cdots + \left\|x_{k+1} - x_k\right\|,$$

by (9) we have

$$\lim_{k\to\infty, k\in K} \left\|x_{k+\ell(k)} - x_k\right\| = 0,$$

which implies

$$\lim_{k\to\infty, k\in K} x_{k+\ell(k)} = \bar{x}.$$

By the instructions of Algorithm TCG, we have for all $k \in K$

$$B_{k+\ell(k)}d_{k+\ell(k)} = -\nabla f\big(x_{k+\ell(k)}\big) + \rho_{k+\ell(k)}, \tag{10}$$

where $\|\rho_{k+\ell(k)}\| \leq \eta_{k+\ell(k)}\|\nabla f(x_{k+\ell(k)})\|$, so that, as $\nabla f(x_{k+\ell(k)}) \to \nabla f(\bar{x})$ and $\eta_{k+\ell(k)} \to 0$,

$$\lim_{k\to\infty, k\in K} \big\|\rho_{k+\ell(k)}\big\| = 0. \tag{11}$$

In the case where

$$\lim_{k\to\infty, k\in K} \lambda_{\min}\big(B_{k+\ell(k)}\big) = 0,$$

as $\nabla f(x_{k+\ell(k)}) \to \nabla f(\bar{x})$, by assumption (b) we have $\nabla f(\bar{x}) = 0$.

Let us assume now that the sequence of minimum eigenvalues $\{\lambda_{\min}(B_{k+\ell(k)})\}$ does not converge to zero, so that there exists an infinite subset $K' \subseteq K$ for which

$$\lambda_{\min}\big(B_{k+\ell(k)}\big) \geq \bar{\lambda} > 0,$$

for all $k \in K'$. Then, from (10) we can write, for all $k \in K'$

$$\begin{aligned}
\frac{\big|\nabla f\big(x_{k+\ell(k)}\big)^T d_{k+\ell(k)}\big|}{\big\|d_{k+\ell(k)}\big\|} &= \frac{\big|d_{k+\ell(k)}^T B_{k+\ell(k)} d_{k+\ell(k)} - \rho_{k+\ell(k)}^T d_{k+\ell(k)}\big|}{\big\|d_{k+\ell(k)}\big\|} \\
&\geq \frac{d_{k+\ell(k)}^T B_{k+\ell(k)} d_{k+\ell(k)} - \big|\rho_{k+\ell(k)}^T d_{k+\ell(k)}\big|}{\big\|d_{k+\ell(k)}\big\|} \\
&\geq \lambda_{\min}\big(B_{k+\ell(k)}\big)\big\|d_{k+\ell(k)}\big\| - \big\|\rho_{k+\ell(k)}\big\| \\
&\geq \bar{\lambda}\big\|d_{k+\ell(k)}\big\| - \big\|\rho_{k+\ell(k)}\big\|,
\end{aligned}$$

and hence, by (8) and (11) $\lim_{k\to\infty, k\in K'} \|d_{k+\ell(k)}\| = 0$. Therefore, from (10) and assumption (a) we have, for all $k \in K'$

$$\big\|\nabla f(x_{k+\ell(k)})\big\| \leq \big\|B_{k+\ell(k)}\big\|\big\|d_{k+\ell(k)}\big\| + \big\|\rho_{k+\ell(k)}\big\| \leq c\big\|d_{k+\ell(k)}\big\| + \big\|\rho_{k+\ell(k)}\big\|,$$

so that, taking limits for $k \to \infty$, $k \in K'$, by (11) we have again $\|\nabla f(\bar{x})\| = 0$. □

Moreover, it is possible to show that, under suitable assumptions and using the Armijo's line search, the convergence rate is superlinear. This result extends that stated in Proposition 1.15 of [1] to the truncation scheme.

Proposition 3. *Let $\{x_k\}$ be the sequence generated by the iterative scheme* (2). *Assume that B_k is symmetric positive semidefinite, $\nabla f(x_k) \in \mathcal{R}(B_k)$, and d_k is computed by means of Algorithm* TCG *where $\eta_k \to 0$ for $k \to \infty$. Moreover, assume that $\{x_k\}$*

converges to $x^\star$, where $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star)$ is positive definite, $\nabla f(x_k) \neq 0$ for all k, and that

$$\lim_{k\to\infty} \frac{\|[B_k^\dagger - \nabla^2 f(x^\star)^{-1}]\nabla f(x_k) - B_k^\dagger \rho_k\|}{\|\nabla f(x_k)\|} = 0, \tag{12}$$

where $B_k^\dagger$ is the pseudoinverse matrix of B_k, and ρ_k is defined in (4)*. Then, if α_k is chosen by means of the Armijo's rule with initial stepsize $\alpha = 1$, we have*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^\star\|}{\|x_k - x^\star\|} = 0.$$

Furthermore, there exists an integer $\bar{k} \geq 0$ such that $\alpha_k = 1$, for all $k \geq \bar{k}$.

Proof: We first prove that there exists a $\bar{k} \geq 0$ such that, for all $k \geq \bar{k}$, we have $\alpha_k = 1$. Since $d_k = -B_k^\dagger[\nabla f(x_k) - \rho_k]$, by the mean value theorem we have

$$\begin{aligned} f(x_k) - f(x_k + d_k) &= -\nabla f(x_k)^T d_k - \frac{1}{2} d_k^T \nabla^2 f(\hat{x}_k)\, d_k \\ &= \nabla f(x_k)^T B_k^\dagger [\nabla f(x_k) - \rho_k] \\ &\quad -\frac{1}{2}[\nabla f(x_k) - \rho_k]^T B_k^\dagger \nabla^2 f(\hat{x}_k) B_k^\dagger [\nabla f(x_k) - \rho_k], \end{aligned} \tag{13}$$

where $\hat{x}_k = x_k + \theta_k d_k$, with $\theta_k \in (0, 1)$. From (12) we have

$$B_k^\dagger[\nabla f(x_k) - \rho_k] \to 0,$$

so that $d_k \to 0$, and $\hat{x}_k$ converges to $x^\star$.

Now, since α_k is determined by the Armijo's rule (see e.g. [1], p. 20), i.e., such that

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k,$$

where $\gamma \in (0, 1/2)$, we have to show that, for k sufficiently large

$$f(x_k) - f(x_k + d_k) \geq -\gamma \nabla f(x_k)^T d_k,$$

i.e., taking into account (13), that

$$\begin{aligned} &\nabla f(x_k)^T B_k^\dagger[\nabla f(x_k) - \rho_k] - \frac{1}{2}[\nabla f(x_k) - \rho_k]^T B_k^\dagger \nabla^2 f(\hat{x}_k) B_k^\dagger [\nabla f(x_k) - \rho_k] \\ &\quad \geq \gamma \nabla f(x_k)^T B_k^\dagger [\nabla f(x_k) - \rho_k], \end{aligned}$$

or equivalently,

$$\begin{aligned} &(1-\gamma)\nabla f(x_k)^T B_k^\dagger \nabla f(x_k) \\ &\geq (1-\gamma)\nabla f(x_k)^T B_k^\dagger \rho_k + \frac{1}{2}[\nabla f(x_k) - \rho_k]^T B_k^\dagger \nabla^2 f(\hat{x}_k) B_k^\dagger [\nabla f(x_k) - \rho_k]. \end{aligned}$$

Then, denoting by $g_k = \nabla f(x_k)/\|\nabla f(x_k)\|$ and $\omega_k = \rho_k/\|\nabla f(x_k)\|$, we have to show that

$$(1-\gamma)g_k^T B_k^\dagger g_k \geq (1-\gamma)g_k^T B_k^\dagger \omega_k + \frac{1}{2}[B_k^\dagger(g_k-\omega_k)]^T \nabla^2 f(\hat{x}_k)[B_k^\dagger(g_k-\omega_k)]. \tag{14}$$

We observe that condition (12) can be written as

$$B_k^\dagger g_k = \nabla^2 f(x^\star)^{-1} g_k + B_k^\dagger \omega_k + \zeta_k, \tag{15}$$

where $\{\zeta_k\}$ denotes a vector sequence with $\zeta_k \to 0$, so that (14) becomes

$$\begin{aligned}(1-\gamma)&g_k^T[\nabla^2 f(x^\star)^{-1} g_k + B_k^\dagger \omega_k + \zeta_k] \\ &\geq (1-\gamma)\omega_k^T[\nabla^2 f(x^\star)^{-1} g_k + B_k^\dagger \omega_k + \zeta_k] \\ &+\frac{1}{2}[\nabla^2 f(x^\star)^{-1} g_k + \zeta_k]^T \nabla^2 f(\hat{x}_k)[\nabla^2 f(x^\star)^{-1} g_k + \zeta_k],\end{aligned}$$

i.e., since $\hat{x}_k \to x^\star$ and $\nabla^2 f(\hat{x}_k) \to \nabla^2 f(x^\star)$,

$$\begin{aligned}\left(\frac{1}{2}-\gamma\right)&g_k^T \nabla^2 f(x^\star)^{-1} g_k \geq (1-\gamma)\omega_k^T \nabla^2 f(x^\star)^{-1} g_k \\ &-(1-\gamma)\,\omega_k^T[B_k^\dagger g_k - B_k^\dagger \omega_k] + \tau_k,\end{aligned} \tag{16}$$

where $\{\tau_k\}$ is some scalar sequence with $\tau_k \to 0$. Now, we have

$$\omega_k^T[B_k^\dagger g_k - B_k^\dagger \omega_k] = \omega_k^T[(B_k^\dagger - \nabla^2 f(x^\star)^{-1})g_k - B_k^\dagger \omega_k] + \omega_k^T \nabla^2 f(x^\star)^{-1} g_k,$$

so that, since $\omega_k \to 0$ and $\|g_k\| = 1$, by (12) $\omega_k^T[B_k^\dagger g_k - B_k^\dagger \omega_k] \to 0$. Then, since $(1/2) - \gamma > 0$ and $\nabla^2 f(x^\star)$ is positive definite, inequality (16) is satisfied for k sufficiently large. To show superlinear convergence we write, for $k \geq \bar{k}$,

$$x_{k+1} - x^\star = x_k - x^\star - B_k^\dagger \nabla f(x_k) + B_k^\dagger \rho_k. \tag{17}$$

From (15) we have

$$B_k^\dagger \nabla f(x_k) = \nabla^2 f(x^\star)^{-1} \nabla f(x_k) + B_k^\dagger \rho_k + \|\nabla f(x_k)\|\zeta_k. \tag{18}$$

Since $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star)$ is positive definite, from Taylor's theorem we obtain

$$\nabla f(x_k) = \nabla^2 f(x^\star)(x_k - x^\star) + o(\|x_k - x^\star\|)$$

and

$$\begin{aligned}\nabla^2 f(x^\star)^{-1} \nabla f(x_k) &= x_k - x^\star + o(\|x_k - x^\star\|) \\ \|\nabla f(x_k)\| &= O(\|x_k - x^\star\|).\end{aligned}$$

Using these two relations in (18) and considering that

$$\zeta_k \frac{O(\|x_k - x^\star\|)}{\|x_k - x^\star\|} \to 0,$$

we obtain

$$B_k^\dagger \nabla f(x_k) = x_k - x^\star + o(\|x_k - x^\star\|) + B_k^\dagger \rho_k$$

and (17) becomes

$$x_{k+1} - x^\star = o(\|x_k - x^\star\|),$$

i.e.,

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^\star\|}{\|x_k - x^\star\|} = 0.$$

□

3. A truncated nonmonotone Gauss-Newton method

We refer to the version of the Gauss-Newton method described in Section 2 of [9]. It is based on the Ben-Israel iteration $x_{k+1} = x_k + d_k^{(m)}$, where $d_k^{(m)}$ is the minimum-norm solution of the linear least-squares problem $\min_d \|J(x_k)d + r(x_k)\|^2$. Obviously, in a truncated version of it the search direction d_k will be an approximation of $d_k^{(m)}$, computed by means of Algorithm TCG. To ensure the global convergence it is necessary to use a line search technique for computing the stepsize along d_k. Moreover, as discussed in [9], it is necessary to impose suitable conditions on d_k, at least at a subsequence of iterates. In particular [9], we take as search direction the solution of the equation $[J(x_k)^T J(x_k) + D_k]d = -J(x_k)^T r(x_k)$, where D_k is a diagonal matrix suitably chosen to ensure that d_k is gradient-related. Even here the use of Algorithm TCG gives an approximate solution. As regards the line search technique, we adopt that employed in [9], which allowed us to obtain the global convergence together with the superlinear rate.

Nonmonotone Line Search Algorithm (NLS)

Data. $\gamma \in (0, 1)$, $0 < \sigma_1 < \sigma_2 < 1$, integer $M > 1$.
Step 0. Set $\alpha = 1$.
Step 1. If

$$f(x_k + \alpha d_k) \le \max_{0\le j\le \min(k,M)} [f(x_{k-j})] - \gamma\alpha^2 \|d_k\|^3,$$

set $\alpha_k = \alpha$ and stop.
Step 2. Choose $\sigma \in [\sigma_1, \sigma_2]$, set $\alpha = \sigma\alpha$ and go to Step 1.

Assuming that d_k is a descent direction for f at x_k, it can be shown that Algorithm NLS is well-defined [9]. In the sequel we prove the convergence properties of the following algorithmic scheme.

Truncated Nonmonotone Gauss-Newton (TNMGN) Stabilization Algorithm

Data. $x_0 \in R^n$, $\varepsilon > 0$, integer $p > 1$.
Step 0. Set $k = 0$, $i = 1$.
Step 1. Compute $J(x_k)$ and $\nabla f(x_k) = J(x_k)^T r(x_k)$, and verify the stopping criterion.
Step 2. If $i < p$, compute by Algorithm TCG an approximate solution d_k of

$$J(x_k)^T J(x_k)\, d \;=\; -J(x_k)^T r(x_k),$$

set $i = i + 1$, and go to Step 4.
Step 3. Compute $D_k = \min\{\varepsilon, \|\nabla f(x_k)\|\} I$, and compute by Algorithm TCG an approximate solution d_k of

$$[J(x_k)^T J(x_k) + D_k]d \;=\; -J(x_k)^T r(x_k).$$

Set $i = 1$.
Step 4. Compute by Algorithm NLS the stepsize α_k. Set $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$, and go to Step 1.

Note that the search direction d_k computed at Step 2 or Step 3 is of descent for f by (iii) of Proposition 1, whatever the stopping criterion of TCG may be.

Proposition 4. *Let $\{x_k\}$ be the sequence generated by Algorithm* TNMGN*, and assume that in Algorithm* TCG *$\eta_k \to 0$ for $k \to \infty$. Moreover, assume that the level set $\Omega_o = \{x \in R^n : f(x) \le f(x_o)\}$ is compact. Then,*

(i) *the sequence $\{f(x_k)\}$ converges*;
(ii) $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$;
(iii) *every limit point of $\{x_k\}$ is a stationary point of $f(x)$.*

Proof: The same proof of Proposition 3.1 in [9] can be used for proving (i) and (ii). In particular, note that, since $f(x_k) \le f(x_o)$ for all k, we have $\{x_k\} \subset \Omega_o$.

To prove (iii), let K_p be the subset of iterates where Step 3 is performed, i.e., $K_p = \{p - 1, 2p - 1, \ldots, jp - 1, \ldots\}$, and denote by

$$B_k = \begin{cases} J(x_k)^T J(x_k), & \text{for } k \notin K_p \\ J(x_k)^T J(x_k) + D_k, & \text{for } k \in K_p. \end{cases} \tag{19}$$

We note that

$$\lambda_{\min}(B_k) = \lambda_{\min}(J(x_k)^T J(x_k)) + \min\{\varepsilon, \|\nabla f(x_k)\|\}, \quad \text{for } k \in K_p$$

and

$$\lambda_{\max}(B_k) \leq \lambda_{\max}(J(x_k)^T J(x_k)) + \min\{\varepsilon, \|\nabla f(x_k)\|\}, \quad \text{for all } k,$$

so that, by the continuity of $\nabla f(x)$, and taking into account that $\{x_k\} \subset \Omega_o$, the sequence $\{B_k\}$ satisfies assumptions (a) and (b).

We show that condition (8) holds, i.e.,

$$\lim_{k\to\infty, k\in K_p} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0,$$

where d_k is the solution of

$$B_k d = -\nabla f(x_k) + \rho_k, \tag{20}$$

and $\|\rho_k\| \leq \eta_k \|\nabla f(x_k)\|$. By contradiction, we assume that there exists an infinite subset $K \subseteq K_p$ such that $\{x_k\}_{k\in K}$ and $\{d_k/\|d_k\|\}_{k\in K}$ converge to some $\hat{x}$ and $\hat{d}$ respectively, and

$$\lim_{k\to\infty, k\in K} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\hat{x})^T \hat{d} < 0. \tag{21}$$

If there exists a subset $K' \subseteq K$ such that $\|d_k\| \to 0$ for $k \to \infty$, $k \in K'$, since B_k satisfies assumption (a) and $\rho_k \to 0$, from (20) we have $\|\nabla f(x_k)\| \to 0$, so that $\nabla f(\hat{x}) = 0$, which contradicts (21). Therefore, by (ii), it follows that $\alpha_k \to 0$ for $k \to \infty$, $k \in K$. Then, by the instructions of Algorithm NLS we have, for $k \in K$ sufficiently large,

$$f\left(x_k + \frac{\alpha_k}{\sigma_k} d_k\right) > f(x_k) - \gamma \frac{\alpha_k^2}{\sigma_k^2} \|d_k\|^3,$$

where $\sigma_k \in [\sigma_1, \sigma_2] \subset (0, 1)$, and by the mean value theorem, it follows that

$$\nabla f\left(x_k + \theta_k \frac{\alpha_k}{\sigma_k} d_k\right)^T d_k/\|d_k\| > -\gamma \left(\frac{\alpha_k}{\sigma_k}\right) \|d_k\|^2, \qquad \theta_k \in (0, 1). \tag{22}$$

We observe that, for $k \in K_p$

$$\begin{aligned} \|d_k\| &\leq \left\|B_k^{-1}\right\| \left\|\rho_k - \nabla f(x_k)\right\| \\ &\leq (1 + \eta_k) \left\|B_k^{-1}\right\| \left\|\nabla f(x_k)\right\|, \end{aligned}$$

and since by $\lim_{k\to\infty, k\in K} \|\nabla f(x_k)\| = \|\nabla f(\hat{x})\| \neq 0$, $J(x_k)^T J(x_k) + D_k$ is uniformly positive definite for $k \in K$, the subsequence $\{\|d_k\|\}_{k\in K}$ is bounded. Therefore, in (22) taking limits for $k \to \infty$, $k \in K$, since $\alpha_k \to 0$, we have

$$\nabla f(\hat{x})^T \hat{d} \geq 0,$$

which contradicts (21). Hence, from Proposition 2, (iii) is proved. □

Proposition 5. *Let $\{x_k\}$ be the sequence generated by Algorithm* TNMGN*, and assume that $\eta_k \to 0$ for $k \to \infty$. Moreover, assume that $\{x_k\}$ converges to $x^\star$, where $f(x^\star) = 0$, $\nabla f(x^\star) = 0$, and $\nabla^2 f(x^\star)$ is positive definite. Then, there exists an integer $\bar{k} \geq 0$ such that, for all $k \geq \bar{k}$, $\alpha_k = 1$, and the sequence $\{x_k\}$ converges superlinearly.*

Proof: As $f(x^\star) = 0$ (the zero-residual case), we have $\nabla^2 f(x^\star) = J(x^\star)^T J(x^\star)$. Since $\lim_{k\to\infty} \nabla f(x_k) = 0$, then $D_k \to 0$ for $k \to \infty$, $k \in K_p$, so that, for the matrix B_k in (19), we have $\lim_{k\to\infty}(B_k^\dagger - \nabla^2 f(x^\star)^{-1}) = 0$, and hence (12) is satisfied. Therefore, employing the same arguments used in the proof of Proposition 3.2 in [9] and Proposition 3 above, the superlinear convergence rate is proved. □

4. Numerical results

As regards the implementative aspects of Algorithm TNMGN, we adopt in Algorithm TCG the simple truncation rule given by

$$\eta_k = 10^{-1} \min\{1/(k+1), \|\nabla f(x_k)\|\},$$

as suggested in [12] in connection with a truncated-Newton method. Moreover, as discussed in [9], we try to modify the matrix $J(x_k)^T J(x_k)$ more than every p iterations. Namely, we use here a rule based on the stepsize for establishing whether the matrix is modified more frequently. Specifically, the matrix will be modified whenever the unit stepsize along the Gauss-Newton search direction was rejected at the previous iteration, and in any case after $p - 1$ iterations without matrix modification.

The numerical results reported below have been obtained by Algorithm TNMGN with the following implementative choices [9]:

- $p = 20$;
- stopping criterion: $\|\nabla f(x_k)\| \leq 10^{-6}$ or $f(x_k) \leq 10^{-8}$;
- line search parameters: $\gamma = 10^{-4}$, $M = 10$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, and at Step 2 the scalar σ is computed by means of a quadratic interpolation formula.

We have considered a set of standard zero-residual large-scale test problems from the literature (the Penalty I function represents a small-residual problem). In particular, functions 1–7 are taken from [11], function 8 from [2], and functions 9–20 from [10]. The following list specifies the functions considered.

1) Extended Rosenbrock, No. 21 in [11]
2) Extended Powell singular, No. 22
3) Penalty I, No. 23
4) Variably dimensioned, No. 25
5) Trigonometric, No. 26
6) Broyden tridiagonal, No. 30
7) Broyden banded, No. 31
8) No. 3 in [2]
9) Generalized Broyden tridiagonal, No. 32 in [10]

10) Tridiagonal system, No. 45
11) Structured Jacobian, No. 46
12) Trigonometric - exponential system, No. 52
13) Singular Broyden, No. 54
14) Five-diagonal system, No. 55
15) Extended Freudenstein & Roth, No. 57
16) Extended Wood, No. 61
17) Tridiagonal exponential, No. 62
18) No. 75
19) No. 76
20) No. 77

The computational advantage deriving from the use of a truncation strategy was already assessed (see, e.g., [2, 4, 7, 12]). We show here the advantage amount in the solution of large-scale problems. In Table 1 we compare the results obtained by Algorithm TNMGN with those obtained by the same algorithm without the truncation scheme (NMGN), i.e., by taking in Algorithm TCG η_k fixed at a low value. For a correct

Table 1. Comparison of the results obtained with Algorithm TNMGN, with Algorithm NMGN (the non-truncated version), and with Algorithm TN.

	TNMGN			NMGN			TN		
Function	$n_i(=n_g)$	n_f	n_{cg}	$n_i(=n_g)$	n_f	n_{cg}	n_i	$n_f(=n_g)$	n_{cg}
1)	13	15	32	8	11	21	16	40	42
2)	13	13	69	12	12	445	12	27	40
3)	131	205	364	194	315	4142	1842	1885	3686
4)	23	23	44	23	23	58	12	28	22
5)	10	11	51	11	22	4031	65	367	780
6)	6	6	26	4	4	65	14	18	83
7)	7	7	19	6	6	53	14	22	56
8)	13	13	256	13	13	584	1579	1583	3281
9)	7	8	26	6	7	101	15	23	54
10)	23	23	327	17	18	690		(*)	
11)	10	10	100	5	5	146	31	54	525
12)	12	17	55	7	9	59	23	34	83
13)	11	11	98	18	22	15066	19	66	64
14)	29	31	943	14	15	1670	77	208	1066
15)	13	13	27	12	12	35		(*)	
16)	22	23	77	25	29	148	158	414	488
17)	3	3	4	3	3	5	5	6	10
18)	981	2434	365395	969	2526	569369	1878	9978	33402
19)	5	5	8	5	5	8	9	10	18
20)	5	5	9	5	5	9	4	6	8

(*) Solution not found within 10^4 function evaluations.

Table 2. Comparison of the results obtained with Algorithm TNMGN and with the E04DGF routine.

	TNMGN			E04DGF		
Function	$n_i(=n_g)$	n_f	time (sec.)	n_i	$n_f\ (=n_g)$	time (sec.)
1)	13	15	0.008	22	96	0.055
2)	13	13	0.008	73	187	0.141
3)	131	205	0.066	2734	6650	9.961
4)	23	23	0.012	11	48	0.031
5)	10	11	4.160	60	122	120.309
6)	6	6	0.004	57	128	0.117
7)	7	7	0.020	61	228	0.285
8)	13	13	0.027	204	443	0.379
9)	7	8	0.008	38	84	0.078
10)	23	23	0.043	230	1878	0.676
11)	10	10	0.023	5	35	0.063
12)	12	17	0.059	55	134	0.305
13)	11	11	0.016	32	238	0.137
14)	29	31	0.125	256	2085	1.344
15)	13	13	0.008	10	28	0.023
16)	22	23	0.020	89	297	0.801
17)	3	3	0.012	2	5	0.020
18)	981	2434	82.781	27009	55533	133.434
19)	5	5	0.004	3	8	0.012
20)	5	5	3.020	5	13	2.164

comparison, we take $\eta_k = 10^{-7}$, according to the final value of the truncation rule. Moreover, in the same table we report for comparison the results obtained with the code TN downloaded from the URL http://iris.gmu.edu/~snash/nash/software/, which is an implementation of a truncated-Newton method suggested in [12]. For each problem we report the numbers n_i of iterations and n_f of function evaluations required for satisfying the stopping criterion (the gradient is evaluated only once at each iteration by Algorithms TNMGN and NMGN, so that $n_g = n_i$). Moreover, we report the total number n_{cg} of iterations performed by the conjugate gradient algorithm. In all problems, the dimension is $n = 1000$ ($m = n$, apart from function 3, the Penalty I function, where $m = n + 1$, and function 4, the Variably dimensioned function, where $m = n + 2$). We remark that the behaviour of the algorithms does not change significantly for higher dimensions, because the structure of the problems remains substantially unchanged.

We observe that the performance of the two versions of the algorithm is similar in terms of n_i and n_f, which suggests the effectiveness of the search direction d_k computed by TCG. However, as expected, the number n_{cg} is lower in the truncated version. In particular, in four problems over twenty, n_{cg} is smaller than 20% of that in the non-truncated version, in five problems it is smaller than 50%, and in one problem only it is slightly greater. As regards the results obtained with the code TN, the solution was not

found within 10^4 function evaluations for functions 10 and 15. In the remaining eighteen problems, in five of them the performance of Algorithm TNMGN is clearly superior, in eight is better, and in five is worse. We note also that in function 5 the results of Algorithm TN were obtained by setting the stopping criterion on the gradient norm at 10^{-5}, due to the occurrence of line search failure at 10^{-6}.

Finally, in Table 2 we compare the results obtained by Algorithm TNMGN with those obtained by the E04DGF routine of the NAG library, which is the implementation of a limited-memory quasi-Newton method, the only one suggested for large-scale problems. Note that the routine E04GBF, specifically designed for nonlinear least-squares, is not suitable for large-scale problems. In fact, since it uses the singular-value decomposition of the Jacobian matrix to calculate the search direction [6], a very high computation time is required. For comparison we report, together with n_i, n_f and n_g ($n_f = n_g$ for E04DGF routine) , the CPU time in seconds. We observe that, in thirteen problems over twenty, the performance of Algorithm TNMGN is clearly superior. In the remaining seven problems, although in some of them the number of iterations performed by Algorithm TNMGN is larger, the numbers n_f and n_g are always smaller, and only in the last problem the time is slightly longer. From these results it appears that the method proposed here may be a valuable alternative for solving large-scale least-squares problems.

Acknowledgment

The authors are grateful to Prof. Stefano Lucidi, University of Rome "La Sapienza", for his careful reading of the manuscript and for providing useful suggestions.

References

1. D.P. Bertsekas, Constrained Optimization and Lagrange Multiplier Methods, Academic Press: New York, NY, 1980.
2. H. Dan, N. Yamashita, and M. Fukushima, "Convergence properties of the inexact Levenberg-Marquardt method under local error bound," Optimization Methods and Software, vol. 17, pp. 605–626, 2002.
3. R.S. Dembo, S.C. Eisenstat, and T. Steihaug, "Inexact Newton methods," SIAM Journal on Numerical Analysis, vol. 19, pp. 400–408, 1982.
4. R.S. Dembo and T. Steihaug, "Truncated-Newton algorithms for large-scale unconstrained optimization," Mathematical Programming, vol. 26, pp. 190–212, 1983.
5. J.E. Dennis Jr. and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Inc., Englewood Cliffs: New Jersey, 1983.
6. P.E. Gill and W. Murray, "Algorithms for the solution of the nonlinear least-squares problems," SIAM Journal on Numerical Analysis, vol. 15, pp. 977–992, 1978.
7. L. Grippo, F. Lampariello and S. Lucidi, "A truncated Newton method with nonmonotone line search for unconstrained optimization," Journal of Optimization Theory and Applications, vol. 60, pp. 401–419, 1989.
8. M.R. Hestenes, Conjugate Direction Methods in Optimization, Springer Verlag: New York, 1980.
9. F. Lampariello and M. Sciandrone, "Use of the minimum-norm search direction in a nonmonotone version of the Gauss-Newton method," Journal of Optimization Theory and Applications, vol. 119, pp. 65–82, 2003.
10. L. Lukšan and J. Vlček, "Test Problems for Unconstrained Optimization," Academy of Sciences of the Czech Republic, Institute of Computer Science, Technical Report no. 897, November 2003.

11. J.J. Moré, B.S. Garbow, and K.E. Hillstrom, "Testing unconstrained optimization software," ACM Trans. Math. Software, vol. 7, pp. 17–41, 1981.
12. S.G. Nash, "A survey of truncated-Newton methods," Journal of Computational and Applied Mathematics, vol. 124, pp. 45–59, 2000.
13. J. Nocedal and S.J. Wright, Numerical Optimization, Springer Series in Operations Research: Springer-Verlag, 1999.